/\*

\* motor\_control.c

\*

\* Created on: Jan 26, 2023

\* Author: udin

\*/

#include "Handler.h"

#include "motor\_control.h"

/\* bldc \*/

esc bldc\_1 = {.htim = &htim12, .channel = TIM\_CHANNEL\_1, .min\_us = 1100, .max\_us = 1900};

esc bldc\_2 = {.htim = &htim12, .channel = TIM\_CHANNEL\_2, .min\_us = 1100, .max\_us = 1900};

/\* omni & mechanism \*/

motor\_channel LeftFront = {

.in1\_ = &htim3, .in2\_ = &htim3,

.ch1\_ = TIM\_CHANNEL\_1, .ch2\_ = TIM\_CHANNEL\_4,

//.en\_port = ENA\_L\_FRONT\_GPIO\_Port,

//.en\_pin = ENA\_L\_FRONT\_Pin

.en\_port = DRV\_ENABLE\_L\_FRONT\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_L\_FRONT\_Pin

};

motor\_channel LeftBack = {

.in1\_ = &htim1, .in2\_ = &htim1,

.ch1\_ = TIM\_CHANNEL\_1, .ch2\_ = TIM\_CHANNEL\_2,

//.en\_port = ENA\_L\_BACK\_GPIO\_Port,

//.en\_pin = ENA\_L\_BACK\_Pin

.en\_port = DRV\_ENABLE\_L\_BACK\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_L\_BACK\_Pin

};

motor\_channel RightFront = {

.in1\_ = &htim3, .in2\_ = &htim3,

.ch1\_ = TIM\_CHANNEL\_2, .ch2\_ = TIM\_CHANNEL\_3,

//.en\_port = ENA\_R\_FRONT\_GPIO\_Port,

//.en\_pin = ENA\_R\_FRONT\_Pin

.en\_port = DRV\_ENABLE\_R\_FRONT\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_R\_FRONT\_Pin

};

motor\_channel RightBack = {

.in1\_ = &htim1, .in2\_ = &htim1,

.ch1\_ = TIM\_CHANNEL\_4, .ch2\_ = TIM\_CHANNEL\_3,

//.en\_port = ENA\_R\_BACK\_GPIO\_Port,

//.en\_pin = ENA\_R\_BACK\_Pin

.en\_port = DRV\_ENABLE\_R\_BACK\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_R\_BACK\_Pin

};

motor\_channel X\_penembak = {

.in1\_ = &htim4, .in2\_ = &htim4,

.ch1\_ = TIM\_CHANNEL\_2, .ch2\_ = TIM\_CHANNEL\_1,

.en\_port = DRV\_ENABLE\_X\_PENEMBAK\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_X\_PENEMBAK\_Pin

//.en\_port = ENA\_X\_Penembak\_GPIO\_Port,

//.en\_pin = ENA\_X\_Penembak\_Pin

};

motor\_channel Y\_penembak = {

.in1\_ = &htim4, .in2\_ = &htim4,

//.ch1\_ = TIM\_CHANNEL\_3, .ch2\_ = TIM\_CHANNEL\_4,

//kelinci

.ch1\_ = TIM\_CHANNEL\_4, .ch2\_ = TIM\_CHANNEL\_3,

.en\_port = DRV\_ENABLE\_Y\_PENEMBAK\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_Y\_PENEMBAK\_Pin

//.en\_port = DRV\_ENABLE\_L\_FRONT\_GPIO\_Port,

//.en\_pin = DRV\_ENABLE\_L\_FRONT\_Pin

};

motor\_channel Chain\_lift = {

.in1\_ = &htim10, .in2\_ = &htim11,

.ch1\_ = TIM\_CHANNEL\_1, .ch2\_ = TIM\_CHANNEL\_1,

.en\_port = DRV\_ENABLE\_GRIPPER\_MT\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_GRIPPER\_MT\_Pin

};

motor\_channel Chain\_lift2 = {

.in1\_ = &htim13, .in2\_ = &htim14,

.ch1\_ = TIM\_CHANNEL\_1, .ch2\_ = TIM\_CHANNEL\_1,

.en\_port = DRV\_ENABLE\_RSVD\_GPIO\_Port,

.en\_pin = DRV\_ENABLE\_RSVD\_Pin

};

void bldc\_init(esc \*servo){

HAL\_TIM\_PWM\_Start(servo->htim, servo->channel);

float min\_ccr\_ = \_\_HAL\_TIM\_GET\_AUTORELOAD(servo->htim) \* 0.05f;

float max\_ccr\_ = \_\_HAL\_TIM\_GET\_AUTORELOAD(servo->htim) \* 0.1f;

servo->min\_ccr = lin\_interp(servo->min\_us, 1000.0f, 2000.0f, min\_ccr\_, max\_ccr\_);

servo->max\_ccr = lin\_interp(servo->max\_us, 1000.0f, 2000.0f, min\_ccr\_, max\_ccr\_);

\_\_HAL\_TIM\_SET\_COMPARE(servo->htim, servo->channel, servo->min\_ccr);

}

void bldc\_drive(esc \*servo, uint32\_t \*duty){

if(\*duty >= servo->max\_ccr) \*duty = servo->max\_ccr;

else if(\*duty <= servo->min\_ccr) \*duty = servo->min\_ccr;

else \_\_NOP();

\_\_HAL\_TIM\_SET\_COMPARE(servo->htim, servo->channel, \*duty);

}

void bldc\_duty(esc \*servo, float percent){

uint32\_t u32\_duty = (uint32\_t)(lin\_interp(percent, 0.0f, 100.0f, servo->min\_ccr, servo->max\_ccr));

\_\_HAL\_TIM\_SET\_COMPARE(servo->htim, servo->channel, u32\_duty);

}

void motor\_init(motor\_channel \*wheel\_n){

HAL\_GPIO\_WritePin(wheel\_n->en\_port, wheel\_n->en\_pin, GPIO\_PIN\_SET);

HAL\_TIM\_PWM\_Start(wheel\_n->in1\_, wheel\_n->ch1\_);

HAL\_TIM\_PWM\_Start(wheel\_n->in2\_, wheel\_n->ch2\_);

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in1\_, wheel\_n->ch1\_, 0);

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in2\_, wheel\_n->ch2\_, 0);

}

void motor\_drive(motor\_channel \*wheel\_n, int\_fast16\_t rpm){ //rpm = ccr val

int\_fast16\_t dir = rpm;

uint\_fast32\_t rpm\_ = (uint\_fast32\_t)(abs(rpm));

if(dir > 0){

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in1\_, wheel\_n->ch1\_, rpm\_);

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in2\_, wheel\_n->ch2\_, 0);

}

else if(dir < 0){

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in1\_, wheel\_n->ch1\_, 0);

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in2\_, wheel\_n->ch2\_, rpm\_);

}

else{

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in1\_, wheel\_n->ch1\_, 0);

\_\_HAL\_TIM\_SET\_COMPARE(wheel\_n->in2\_, wheel\_n->ch2\_, 0);

}

}

void disable\_motor(motor\_channel \*wheel\_n){

HAL\_GPIO\_WritePin(wheel\_n->en\_port, wheel\_n->en\_pin, GPIO\_PIN\_RESET);

}

void enable\_motor(motor\_channel \*wheel\_n){

HAL\_GPIO\_WritePin(wheel\_n->en\_port, wheel\_n->en\_pin, GPIO\_PIN\_SET);

}